

[illegible]

3

Sy

MT

MT
MT

MT

MT

MT
MTMT
MTMT
MTMT
MT

MT
MT

MT
MT

MT

MT

MT
MT

MI
MT

MT
MT

MT
MT

MT
MT

MT
MT

MT

111

227

MT

MT
MT

MT

MT

M1
M1MT
MT

M1

M1

10

1

10

100

10

1

```
MM      MM      TTTTTTTTTT  HH      HH      HH      HH      AAAAAA      CCCCCCCC      000000      SSSSSSSS
MM      MM      TTTTTTTTTT  HH      HH      HH      HH      AAAAAA      CCCCCCCC      000000      SSSSSSSS
MMMM    MMMM      TT      HH      HH      HH      HH      AA      AA      CC      00      00      SS
MMMM    MMMM      TT      HH      HH      HH      HH      AA      AA      CC      00      00      SS
MM      MM      TT      HH      HH      HH      HH      AA      AA      CC      00      00      SS
MM      MM      TT      HH      HH      HH      HH      AA      AA      CC      00      00      SS
MM      MM      TT      HHHHHHHHHH  HHHHHHHHHH  AA      AA      CC      00      00      SSSSSS
MM      MM      TT      HHHHHHHHHH  HHHHHHHHHH  AA      AA      CC      00      00      SSSSSS
MM      MM      TT      HH      HH      HH      HH      AAAAAAAAAA  CC      00      00      SS
MM      MM      TT      HH      HH      HH      HH      AAAAAAAAAA  CC      00      00      SS
MM      MM      TT      HH      HH      HH      HH      AA      AA      CC      00      00      SS
MM      MM      TT      HH      HH      HH      HH      AA      AA      CC      00      00      SS
MM      MM      TT      HH      HH      HH      HH      AA      AA      CC      00      00      SS
MM      MM      TT      HH      HH      HH      HH      AA      AA      CCCCCCCC  000000      SSSSSSSS
MM      MM      TT      HH      HH      HH      HH      AA      AA      CCCCCCCC  000000      SSSSSSSS
```

```
LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL  IIIIII      SSSSSSSS
LLLLLLLLLL  IIIIII      SSSSSSSS
```

MTH
Syn
D-C
D-R
D-T
D-T
ERR
GEO
HAC
H-1
H-9
H-P
H-P
LON
MTH
MTH
MTH
MTH
MTH
MTH
MTH
MTH
MTH
MTH
RET
TES
TES
X

PSE

_MT

Pha

Ini
Com
Pas
Syn
Pas
Syn
Pse
Cro
Ass

The
643
The
492

(2) 54
(3) 67
(4) 112
(5) 187
(6) 276
(7) 351

HISTORY : Detailed Current Edit History
DECLARATIONS : Declarative Part of Module
MTH\$HACOS - Standard H Floating Arc cosine
MTH\$HACOS_R8 - Special HACOS routine
MTH\$HACOSD - Standard H Floating Arc cosine
MTH\$HACOSD_R8 - Special HACOSD routine

```
0000 1      .TITLE  MTH$HACOS      ; H Floating Point Arc-cosine routine
0000 2      ; (HACOS,HACOSD)
0000 3      .IDENT /1-006/        ; File: MTHHACOS.MAR  EDIT: RNH1006
0000 4      :
0000 5      :*****
0000 6      :
0000 7      :*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8      :*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9      :*  ALL RIGHTS RESERVED.
0000 10     :
0000 11     :*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12     :*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13     :*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14     :*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15     :*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16     :*  TRANSFERRED.
0000 17     :
0000 18     :*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19     :*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20     :*  CORPORATION.
0000 21     :
0000 22     :*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23     :*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24     :
0000 25     :*****
0000 26     :
0000 27     :
0000 28     :
0000 29     : FACILITY: MATH LIBRARY
0000 30     :++
0000 31     : ABSTRACT:
0000 32     :
0000 33     : MTH$HACOS is a function which returns the H Floating point arc-cosine
0000 34     : in radians of its H Floating point argument. The call is standard call-
0000 35     : by-reference.
0000 36     :
0000 37     : MTH$HACOSD is a function which returns the H Floating point arc-cosine
0000 38     : in radians of its H Floating point argument. The call is standard call-
0000 39     : by-reference.
0000 40     :
0000 41     :--
0000 42     :
0000 43     : VERSION: 1
0000 44     :
0000 45     : HISTORY:
0000 46     : AUTHOR:
0000 47     :      John A. Wheeler, 20-Oct-1979: Version 1
0000 48     :
0000 49     : MODIFIED BY:
0000 50     :
0000 51     :
0000 52     :
```

MTH\$HACOS
1-006

J 9
; H Floating Point Arc-cosine routine 16-SEP-1984 01:33:10 VAX/VMS Macro V04-00 Page 2
HISTORY ; Detailed Current Edit History 6-SEP-1984 11:24:29 [MTHRTL.SRC]MTHHACOS.MAR;1 (2)

```
0000 54      .SBTTL  HISTORY ; Detailed Current Edit History
0000 55
0000 56
0000 57 : Edit History for Version 1 of MTH$HACOS
0000 58 :
0000 59 : 1-001 - Adapted from MTH$GACOS version 1-002.  JAW 20-Oct-1979
0000 60 : 1-002 - Added degree entry points.  RNH 22-MAR-1981
0000 61 : 1-003 - Modified computation of 1 - x^2 to avoid loss of significance
0000 62 :           for arguments >=1/2
0000 63 : 1-004 - Change shared external references to G^  RNH 02-Oct-81
0000 64 : 1-005 - Eliminated symbolic short literals.  RNH 15-Oct-81
0000 65 : 1-006 - Changed MULH2 R2, R0 to MULH (SP)+, R0 in MTH$HACOSD.  RNH 20-Jan-82
```



```
0000 67 .SBTTL DECLARATIONS ; Declarative Part of Module
0000 68
0000 69 ;
0000 70 ; INCLUDE FILES: OTSPARAMS.MAR
0000 71 ;
0000 72 ; EXTERNAL SYMBOLS:
0000 73 ;
0000 74
0000 75 .DSABL GBL ; Force undefineds to error
0000 76 .EXTRN MTH$HATAN R8 ; Arctangent routine
0000 77 .EXTRN MTH$HATAN D R8 ; Arctangent routine
0000 78 .EXTRN MTH$HSQRT R8 ; Square root routine
0000 79 .EXTRN MTH$SSIGNAL ; Math signal routine
0000 80 .EXTRN MTH$K_INVARGMAT ; Error code
0000 81
0000 82 ; EQUATED SYMBOLS:
0000 83
0000 84
0000 85 ;
0000 86 ; MACROS: none
0000 87 ;
0000 88 ; PSECT DECLARATIONS:
0000 89
00000000 90 .PSECT _MTH$CODE PIC,SHR, LONG, EXE, NOWRT
0000 91 ; Program section for math routines
0000 92 ;
0000 93 ; OWN STORAGE: none
0000 94 ;
0000 95 ;
0000 96 ; CONSTANTS:
0000 97 ;
0000 98
0000 99 H_PI_OVER 2:
42D1B544 921F4001 0000 100 .LONG ^X921F4001, ^X42D1B544 ; 1.5707963267948966192313216916397514420
01B8C517 898C8469 0008 101 .LONG ^X898C8469, ^X01B8C517
0010 102 H_PI:
42D1B544 921F4002 0010 103 .LONG ^X921F4002, ^X42D1B544 ; 3.1415926535897932384626433832795028841
01B8C517 898C8469 0018 104 .LONG ^X898C8469, ^X01B8C517
0020 105 H_90:
00000000 68004007 0020 106 .LONG ^X68004007, ^X0 ; 90
00000000 00000000 0028 107 .LONG ^X0, ^X0
0030 108 H_180:
00000000 68004008 0030 109 .LONG ^X68004008, ^X0 ; 180
00000000 00000000 0038 110 .LONG ^X0, ^X0
```

```
0040 112 .SBTTL MTH$HACOS - Standard H Floating Arc cosine
0040 113
0040 114
0040 115 :++
0040 116 : FUNCTIONAL DESCRIPTION:
0040 117
0040 118 : HACOS - H Floating point function
0040 119
0040 120 : HACOS(X) is computed as:
0040 121
0040 122 : If X = 0, then HACOS(X) = PI/2.
0040 123 : If X = 1, then HACOS(X) = 0.
0040 124 : If X = -1, then HACOS(X) = PI.
0040 125 : If 0 < X < 1/2, then HACOS(X) = HATAN(SQRT(1-X**2)/X).
0040 126 : If 1/2 < X < 1, then HACOS(X) = HATAN(SQRT((1-X)*(1+X))/X).
0040 127 : If -1/2 < X < 0, then HACOS(X) = HATAN(SQRT(1-X**2)/X) + PI.
0040 128 : If -1 < X < -1/2, then HACOS(X) = HATAN(SQRT((1-X)*(1+X))/X).
0040 129 : If 1 < |X|, error.
0040 130
0040 131 : CALLING SEQUENCE:
0040 132
0040 133 : hacos.wh.v = MTH$HACOS(x.rh.r)
0040 134
0040 135 : -or-
0040 136
0040 137 : CALL MTH$HACOS(hacos.wh.r, x.rh.r)
0040 138
0040 139 : Because an H-floating result cannot be expressed in 64 bits, it is
0040 140 : returned as the first argument, with the input parameter displaced
0040 141 : to the second argument, in accordance with the Procedure Calling
0040 142 : Standard.
0040 143
0040 144 : INPUT PARAMETERS:
0040 145
00000004 0040 146 : LONG = 4 ; Define longword multiplier
00000008 0040 147 : x = 2 * LONG ; Contents of x is the argument
0040 148
0040 149 : IMPLICIT INPUTS: none
0040 150
0040 151 : OUTPUT PARAMETERS:
0040 152
00000004 0040 153 : hacos = 1 * LONG ; hacos is the result
0040 154
0040 155 : VALUE: H Floating arc-cosine of the argument
0040 156
0040 157 : IMPLICIT OUTPUTS: none
0040 158
0040 159 : COMPLETION CODES: none
0040 160
0040 161 : SIDE EFFECTS:
0040 162
0040 163 : Signals: MTH$_INVARGMAT if |X| > 1 with reserved operand in R0/R1 (copied
0040 164 : to the signal mechanism vector CHF$MCH_R0/R1 by LIB$SIGNAL).
0040 165 : Associated message is: "INVALID ARGUMENT". Result is reserved operand -0.0
0040 166 : unless a user supplied (or any) error handler changes CHF$MCH_R0/R1.
0040 167
0040 168 : NOTE: This procedure disables floating point underflow, enables integer
```



```
0040 169 ; overflow.
0040 170 :
0040 171 :---
0040 172 :
0040 173
41FC 0040 174 .ENTRY MTH$HACOS, ^M<IV, R2, R3, R4, R5, R6, R7, R8>
0042 175 ; Standard call-by-reference entry
0042 176 ; Disable DV (and FU), enable IV
0042 177 MTH$FLAG_JACKET ; Flag that this is a jacket procedure in
0042
6D 00000000'GF 9E 0042 MOVAB G^MTH$$JACKET_HND, (FP)
0049 ; set handler address to jacket
0049 ; handler
0049 178 ; Case of an error in routine
0049 179 ; If an error, convert signal to user PC
0049 180 ; And resignal
50 08 BC 70FD 0049 181 MOVH @x(AP), R0 ; R0/R3 = X = @x(AP)
06 10 004E 182 BSBB MTH$HACOS_R8 ; Call special HACOS routine
04 BC 50 7DFD 0050 183 MOVO R0, @hacos(AP) ; Store result in first argument
04 0055 184 RET ; Return to caller
0056 185
```



```
0056 187      .SBTTL MTH$HACOS_R8 - Special HACOS routine
0056 188
0056 189      ; Special HACOS - used by the standard routine and direct JSB call.
0056 190
0056 191      CALLING SEQUENCE:
0056 192      ; save anything needed in R0:R8
0056 193      MOVH      R0      ; Input in R0/R3
0056 194      JSB      MTH$HACOS_R8
0056 195      ; return with result in R0/R3
0056 196
0056 197
0056 198 MTH$HACOS_R8::
0056 199      TSTW      R0      ; Special HACOS routine
0058 200      BNEQ     TEST_FOR_1.0      ; Compare X with 0
005A 201      ;
005A 202      ; X = 0
005A 203
005A 204
005A 205
005A 206      MOVO     H_PI_OVER_2, R0      ; R0/R3 = PI/2
005F 207      RSB
0060 208      ; Return PI/2 if !X! = 0
0060 209
0060 210      ; 0 < !X!
0060 211
0060 212
0060 213 TEST_FOR_1.0:
0060 214      MOVH     R0, -(SP)      ; Save X on stack
0064 215      BICW     #^X8000, R0      ; R0/R3 = !X!
0069 216      CMPL     R0, #1      ; Compare !X! with 1.0
006D 217      BGEQ     GEQ_TO_1.0      ; Branch if !X! >= 1.0
006F 218
006F 219
006F 220      ; 0 < !X! < 1.0
006F 221
006F 222
006F 223      CMPW      #4000, R0      ; Check for possible loss of
0074 224      BGTR      1$      ; significance
0076 225      SUBH3     R0, #1, -(SP)      ; (SP) = 1 - X
007B 226      ADDH2     #1, R0      ; R0/R3 = 1 + X
007F 227      MULH2     (SP)+, R0      ; R0/R3 = 1 - X^2
0083 228      BRB      2$      ; Join main flow
0085 229 1$:      MULH2     R0, R0      ; R0/R3 = X**2
0089 230      SUBH3     R0, #1, R0      ; R0/R3 = 1.0 - X**2
008E 231 2$:      JSB      G^MTH$HSQRT_R8      ; R0/R3 = HSQRT(1-X**2)
0094 232      DIVH2     (SP), R0      ; R0/R3 = HSQRT(1-X**2)/X
0098 233      JSB      G^MTH$HATAN_R8      ; R0/R3 = HATAN(HSQRT(1-X**2)/X)
009E 234      BRB      TEST_SIGN      ; Branch to TEST_SIGN
00A0 235
00A0 236
00A0 237      ; 1 <= !X!
00A0 238
00A0 239
00A0 240 GEQ_TO_1.0:
00A0 241      BGTR      ERROR      ; Branch to ERROR if !X! > 1.0
00A2 242
00A2 243 ;
```

```
00A2 244 ; |X| = 1.0
00A2 245 ;
00A2 246 ;
50 7CFD 00A2 247 CLRH R0 ; R0/R3 = 0
00A5 248 ;
00A5 249 ;
00A5 250 ; Test the sign of X in order to decide whether to add PI to the result
00A5 251 ;
00A5 252 ;
00A5 253 TEST_SIGN:
00A5 254 TSTH (SP) ; Test the sign of X
06 18 00A8 255 BGEQ RETURN ; Branch if X > 0
50 FF61 CF 60FD 00AA 256 ADDH2 H_PI, R0 ; Add PI to R0/R3 if X < 0
5E 10 C0 00B0 257 RETURN: ADDL2 #T6, SP ; Clear stack
00B3 258 RSB ; Return to caller
00B4 259 ;
00B4 260 ;
00B4 261 ; 1 < |X|, error
00B4 262 ;
00B4 263 ;
5E 10 C0 00B4 264 ERROR: ADDL2 #16, SP ; Clear stack
06 18 DD 00B7 265 PUSHL (SP) ; Return PC from JSB routine
7E 00'8F 9A 00B9 266 MOVZBL #MTH$K_INVARGMAT, -(SP) ; Condition value
50 01 0F 79 00BD 267 ASHQ #15, #T, R0 ; R0/R3 = result = reserved operand -0.0
00C1 268 ; goes to signal mechanism vector
00C1 269 ; (CHF$MCH_R0/R1) so error handler
00C1 270 ; can modify the result.
00000000'GF 52 7C 00C1 271 CLRQ R2 ;
02 FB 00C3 272 CALLS #2, G^MTH$$SIGNAL ; Signal error and use real user's PC
00CA 273 ; independent of CALL vs JSB
05 00CA 274 RSB ; Return - R0 restored from CHF$MCH_R0/R1
```



```
00CB 276 .SBTTL MTH$HACOSD - Standard H Floating Arc cosine
00CB 277
00CB 278
00CB 279 :++
00CB 280 : FUNCTIONAL DESCRIPTION:
00CB 281 :
00CB 282 : HACOSD - H Floating point function
00CB 283 :
00CB 284 : HACOSD(X) is computed as:
00CB 285 :
00CB 286 :     If X = 0, then HACOSD(X) = 90.
00CB 287 :     If X = 1, then HACOSD(X) = 0.
00CB 288 :     If X = -1, then HACOSD(X) = 180.
00CB 289 :     If 0 < X < 1, then HACOSD(X) = ATAN(SQRT(1-X**2)/X).
00CB 290 :     If 1/2 < X < 1, then HACOSD(X) = HATAND(SQRT((1-X)*(1+X))/X).
00CB 291 :     If -1/2 < X < 0, then HACOSD(X) = HATAND(SQRT(1-X**2)/X) + 180.
00CB 292 :     If -1 < X < 1/2, then HACOSD(X) = HATAND(SQRT((1-X)*(1+X))/X) + 180.
00CB 293 :     If 1 < |X|, error.
00CB 294 :
00CB 295 : CALLING SEQUENCE:
00CB 296 :
00CB 297 :     hacosd.wh.v = MTH$HACOSD(x.rh.r)
00CB 298 :
00CB 299 :     -or-
00CB 300 :
00CB 301 :     CALL MTH$HACOSD(hacosd.wh.r, x.rh.r)
00CB 302 :
00CB 303 :     Because an H-floating result cannot be expressed in 64 bits, it is
00CB 304 :     returned as the first argument, with the input parameter displaced
00CB 305 :     to the second argument, in accordance with the Procedure Calling
00CB 306 :     Standard.
00CB 307 :
00CB 308 : INPUT PARAMETERS:
00CB 309 :
00000004 00CB 310 :     LONG = 4 ; Define longword multiplier
00000008 00CB 311 :     x = 2 * LONG ; Contents of x is the argument
00CB 312 :
00CB 313 : IMPLICIT INPUTS: none
00CB 314 :
00CB 315 : OUTPUT PARAMETERS:
00CB 316 :
00000004 00CB 317 :     HACOSD = 1 * LONG ; HACOSD is the result
00CB 318 :
00CB 319 :     VALUE: H Floating arc-cosine of the argument
00CB 320 :
00CB 321 : IMPLICIT OUTPUTS: none
00CB 322 :
00CB 323 : COMPLETION CODES: none
00CB 324 :
00CB 325 : SIDE EFFECTS:
00CB 326 :
00CB 327 : Signals: MTH$_INVARGMAT if |X| > 1 with reserved operand in R0/R1 (col80ed
00CB 328 : to the signal mechanism vector CHF$MCH_R0/R1 by LIB$SIGNAL).
00CB 329 : Associated message is: "INVALID ARGUMENT". Result is reserved operand -0.0
00CB 330 : unless a user supplied (or any) error handler changes CHF$_MCH_R0/R1.
00CB 331 :
00CB 332 : NOTE: This procedure disables floating point underflow, enables integer
```

```
00CB 333 ; overflow.
00CB 334 :
00CB 335 :---
00CB 336
00CB 337
41FC 00CB 338 .ENTRY MTH$HACOSD, ^M<IV, R2, R3, R4, R5, R6, R7, R8>
00CD 339 ; Standard call-by-reference entry
00CD 340 ; Disable DV (and FU), enable IV
00CD 341 MTH$FLAG_JACKET ; Flag that this is a jacket procedure in
6D 00000000'GF 9E 00CD MOVAB G^MTH$$JACKET_HND, (FP)
00D4 ; set handler address to jacket
00D4 ; handler
00D4 342 ; Case of an error in routine
00D4 343 ; If an error, convert signal to user PC
00D4 344 ; And resignal
50 08 BC 70FD 00D4 345 MOVH @x(AP), R0 ; R0/R3 = X = @x(AP)
06 10 00D9 346 BSBB MTH$HACOSD_R8 ; Call special HACOSD routine
04 BC 50 7DFD 00DB 347 MOVJ R0, @HACOSD(AP) ; Store result in first argument
04 00E0 348 RET ; Return to caller
00E1 349
```



```
00E1 351      .SBTTL MTH$HACOSD_R8 - Special HACOSD routine
00E1 352
00E1 353      ; Special HACOSD - used by the standard routine and direct JSB call.
00E1 354      ;
00E1 355      ; CALLING SEQUENCE:
00E1 356      ;   save anything needed in R0:R8
00E1 357      ;   MOVH      R0      ; Input in R0/R3
00E1 358      ;   JSB      MTH$HACOSD_R8
00E1 359      ;   return with result in R0/R3
00E1 360      ;
00E1 361
00E1 362 MTH$HACOSD_R8::
00E1 363      TSTW      R0      ; Special HACOSD routine
50 07 B5 00E1 364      BNEQ      D_TEST_FOR_1.0      ; Compare X with 0
00E3 365      ;
00E5 366      ; X = 0
00E5 367      ;
00E5 368      ;
00E5 369      ;
50 FF36 CF 7DFD 00E5 370      MOVO      H_90, R0      ; R0/R3 = 90
00EB 371      RSB      ; Return 90 if !X! = 0
00EC 372      ;
00EC 373      ; 0 < !X!
00EC 374      ;
00EC 375      ;
00EC 376      ;
00EC 377 D_TEST_FOR_1.0:
50 7E 50 70FD 00EC 378      MOVH      R0, -(SP)      ; Save X on stack
00F0 379      BICW      #X8000, R0      ; R0/R3 = !X!
50 08 50 71FD 00F5 380      CMPL      R0, #1      ; Compare !X! with 1.0
00F9 381      BGEQ      D_GEQ_TO_1.0      ; Branch if !X! >= 1.0
00FB 382      ;
00FB 383      ; 0 < !X! < 1.0
00FB 384      ;
00FB 385      ;
00FB 386      ;
50 0FA0 8F B1 00FB 387      CMPW      #4000, R0      ; Check for possible loss of
00FB 388      BGTR      1$      ; significance
7E 08 50 63FD 0102 389      SUBH3      R0, #1, -(SP)      ; (SP) = 1 - X
50 08 50 60FD 0107 390      ADDH2      #1, R0      ; R0/R3 = 1 + X
50 08 50 64FD 010B 391      MULH2      (SP)+, R0      ; R0/R3 = 1 - X^2
50 09 11 010F 392      BRB      2$      ; Join main flow
50 50 50 64FD 0111 393 1$:      MULH2      R0, R0      ; R0/R3 = X**2
50 08 50 63FD 0115 394      SUBH3      R0, #1, R0      ; R0/R3 = 1.0 - X**2
00000000'GF 16 011A 395 2$:      JSB      G^MTH$HSQRT_R8      ; R0/R3 = HSQRT(1-X**2)
50 50 6E 66FD 0120 396      DIVH2      (SP), R0      ; R0/R3 = HSQRT(1-X**2)/X
00000000'GF 16 0124 397      JSB      G^MTH$HATAND_R8      ; R0/R3 = HATAND(HSQRT(1-X**2)/X)
00FB 398      BRB      D_TEST_SIGN      ; Branch to D_TEST_SIGN
012C 399      ;
012C 400      ; 1 <= !X!
012C 401      ;
012C 402      ;
012C 403      ;
012C 404 D_GEQ_TO_1.0:
00FB 405      BEQL      10$      ; Branch to ERROR if !X! > 1.0
FF83 31 012E 406      BRW      ERROR
0131 407
```

```
0131 408 :  
0131 409 : |X| = 1.0  
0131 410 :  
0131 411 :  
50 7CFD 0131 412 10$: CLRH R0 ; R0/R3 = 0  
0134 413 :  
0134 414 :  
0134 415 : Test the sign of X in order to decide whether to add 180 to the result  
0134 416 :  
0134 417 :  
0134 418 D_TEST_SIGN:  
6E 73FD 0134 419 TSTH (SP) ; Test the sign of X  
06 18 0137 420 BGEQ D_RETURN ; Branch if X > 0  
50 FEF2 CF 60FD 0139 421 ADDH2 H_180, R0 ; Add 180 to R0/R3 if X < 0  
013F 422 D_RETURN:  
5E 10 C0 013F 423 ADDL2 #16, SP ; Clear stack  
05 0142 424 RSB ; Return to caller  
0143 425 :  
0143 426 :  
0143 427 : 1 < |X|, error  
0143 428 :  
0143 429 :  
0143 430 :  
0143 431 :  
0143 432 .END
```


MTH\$HACOS
Symbol table

G 10
; H Floating Point Arc-cosine routine

16-SEP-1984 01:33:10
6-SEP-1984 11:24:29

VAX/VMS Macro V04-00
[MTHRTL.SRC]MTHHACOS.MAR;1

Page 12
(7)

```
D_GEQ_TO_1.0 0000012C R 01
D_RETURN      0000013F R 01
D_TEST_FOR_1.0 000000EC R 01
D_TEST_SIGN   00000134 R 01
ERROR         000000B4 R 01
GEQ_TO_1.0    000000A0 R 01
HACOS         = 00000004
HACOSD        = 00000004
H_180         00000030 R 01
H_90          00000020 R 01
H_PI          00000010 R 01
H_PI_OVER_2   00000000 R 01
LONG          = 00000004
MTH$$JACKET_HND ***** X 01
MTH$$SIGNAL    ***** X 00
MTH$HACOS      00000040 RG 01
MTH$HACOSD     000000CB RG 01
MTH$HACOSD R8  000000E1 RG 01
MTH$HACOS R8   00000056 RG 01
MTH$HATAN R8   ***** X 00
MTH$HATAN R8   ***** X 00
MTH$HSQRT R8   ***** X 00
MTH$K_INVARGMAT ***** X 00
RETURN        000000B0 R 01
TEST_FOR_1.0   00000060 R 01
TEST_SIGN      000000A5 R 01
X              = 00000008
```

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes												
ABS	00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE		
MTH\$CODE	00000143 (323.)	01 (1.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC	LONG		

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	31	00:00:00.08	00:00:00.45
Command processing	110	00:00:00.66	00:00:03.74
Pass 1	95	00:00:01.18	00:00:04.03
Symbol table sort	0	00:00:00.01	00:00:00.01
Pass 2	86	00:00:00.95	00:00:04.49
Symbol table output	4	00:00:00.03	00:00:00.03
Psect synopsis output	2	00:00:00.01	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	330	00:00:02.93	00:00:12.77

The working set limit was 1050 pages.
6438 bytes (13 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 27 non-local and 5 local symbols.
492 source lines were read in Pass 1, producing 14 object records in Pass 2.

1 page of virtual memory was used to define 1 macro.

! Macro library statistics !

Macro library name	Macros defined
-----	-----
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LISS:MTHHACOS/OBJ=OBJ\$:MTHHACOS MSRC\$:MTHJACKET/UPDATE=(ENH\$:MTHJACKET)+MSRC

MTH
Syn

D-C
D-F
ERR
GEO
HAS
HAS
H-S
H-F
LON
MTH
MTH
MTH
MTH
MTH
MTH
MTH
MTH
MTH
MTH
RET
X

PSE

-M1

Pha

Ini
Con
Pas
Syn
Pas
Syn
Pse
Cro
Ass

The
541
The
429
1

0261 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY